1.

The statement in the body of a while loop acts as a decision maker.

**B) False**

2.

An infinite loop may execute indefinitely.

**A) True**

3.

The number of iterations of a counter-controlled loop is known in advance.

**A) True**

4.

Assume all variables are properly declared. The output of the following C++ code is 2 3 4 5.

n = 1;

while (n < 5)

{

n++;

cout << n << " ";

}

**A) True**

5.

In the case of the sentinel-controlled while loop, the first item is read before the while loop is entered.

**A) True**

6.

The control variable in a flag-controlled while loop is a bool variable.

**A) True**

7.

The function eof returns true if the program has read past the end of the input file.

**A) True**

8.

A for loop is typically called an indexed for loop.

**A) True**

9.

Assume all variables are properly declared. The following for loop executes 20 times.

for (i = 0; i <= 20; i++)

cout << i;

**B) False**

10.

A syntax error will result if the control statements of a for loop are omitted.

**B)false**

11.

A do...while loop is a post-test loop.

**A) True**

12.

Both while and for loops are pre-test loops.

**A) True**

13.

The execution of a break statement in a while loop terminates the loop.

**A) True**

14.

After a break statement executes, the program continues to execute with the first statement after the structure.

**A) True**

15.

In \_\_\_\_ structures, the computer repeats particular statements a certain number of times depending on some condition(s).

**A) looping**

16.

What is the output of the following C++ code?

num = 10;

while (num > 10)

num = num - 2;

cout << num << endl;

**D) 10**

17.

Assume all variables are properly declared. What is the output of the following C++ code?

num = 100;

while (num <= 150)

num = num + 5;

cout << num << endl;

**B) 155**

18.

Suppose sum and num are int variables, and the input is 18 25 61 6 -1. What is the output of the following code?

sum = 0;

cin >> num;

while (num != -1)

{

sum = sum + num;

cin >> num;

}

cout << sum << endl;

**C) 110**

19.

Consider the following code. (Assume that all variables are properly declared.)

cin >> ch;

while (cin)

{

cout << ch;

cin >> ch;

}

This code is an example of a(n) \_\_\_\_ while loop.

**C) EOF-controlled**

20.

Which of the following is the initial statement in the following for loop? (Assume that all variables are properly declared.)

int i;

for (i = 1; i < 20; i++)

cout << "Hello World";

cout << "!" << endl;

**A)**

21.

Suppose sum, num, and j are int variables, and the input is 4 7 12 9 -1. What is the output of the following code?

cin >> sum;

cin >> num;

for (j = 1; j <= 3; j++)

{

cin >> num;

sum = sum + num;

}

cout << sum << endl;

**A) 24**

22.

Suppose j, sum, and num are int variables, and the input is 26 34 61 4 -1. What is the output of the code?

sum = 0;

cin >> num;

for (int j = 1; j <= 4; j++)

{

sum = sum + num;

cin >> num;

}

cout << sum << endl;

**B) 125**

23.

Which of the following is a repetition structure in C++?

**D) do...while**

24.

Which executes first in a do...while loop?

**A) the statement**

25.

What is the value of x after the following statements execute?

int x = 5;

int y = 30;

do

x = x \* 2;

while (x < y);

**D) 40**

26.

To use the predefined function abs, the program must include the header file ctype.

**B) False**

27.

To use a predefined function, the program must include the appropriate header file.

**A) True**

28.

Once you write and properly debug a function, you can use it in the program (or different programs) again and again without having to rewrite the same code repeatedly.

**A) True**

29.

A function definition consists of the function heading and the body of the function.

**A) True**

30.

A function’s formal parameter list cannot be empty.

**B) False**

31.

The following function heading in a C++ program is valid:

int funcExp(int u, char v, float g)

**A) True**

32.

The execution of a return statement in a user-defined function terminates the program.

**B) False**

33.

In C++, a function prototype is the function heading without the body of the function.

**A) True**

34.

A function prototype ends with a semicolon.

1. **True**

35.

The return statement return x + 1; first returns the value of x and then increments the value of x.

1. **false**

36.

A value-returning function can return two values via the return statement.

**b) False**

37.

The execution of a C++ program always begins with the function main.

**B) False**

38.

The standard header file for the abs(x)function is \_\_\_\_.

**A)**

39.

The output of the statement:

cout << tolower('$') << endl;

is \_\_\_\_.

**A) '$'**

40.

Functions that do not have a return type are called \_\_\_\_ functions.

**C) void**

41.

Given the following function prototype: int test(float, char); which of the following statements is valid?

**C) int u = test(5.0, '\*');**

42.

A variable or expression listed in a call to a function is called the \_\_\_\_.

**B) actual parameter**

43.

Given the following function:

int next(int x)

{

return (x + 1);

}

what is the output of the following statement?

cout << next(next(5)) << endl;

**C) 7**

44.

Which statement below about prototypes and headers is true?

**B) Prototypes end with a semicolon, but headers do not.**

45.

Which of the following function prototypes is valid?

**D) int funcTest(int, int, float);**

46.

Given the following function prototype: int myFunc(int, int); which of the following statements is valid? Assume that all variables are properly declared.

**B) cout << myFunc(myFunc(7, 8), 15);**

47.

The statement: return 8, 10; returns the value \_\_\_\_.

**B) 10**

48.

The statement: return 37, y, 2 \* 3; returns the value \_\_\_\_.

**D) 6**

49.

Given the following function:

int strange(int x, int y)

{

if (x > y)

return x + y;

else

return x – y;

}

what is the output of the following statement?

cout << strange(4, 5) << endl;

**a)-1**

50.

Given the following function:

int strange(int x, int y)

{

if (x > y)

return x + y;

else

return x – y;

}

what is the output of the following statement?

cout << strange(4, 5) << endl;

**A) -1**